**Title: Real-Time Distance Monitoring and Motor Speed Control via MATLAB and Arduino Interfacing**

**Introduction**

**This project presents a real-time sliding door automation system using MATLAB and Arduino. An ultrasonic sensor detects human presence within 50 cm, and based on the distance, MATLAB sends control signals to Arduino to operate a DC motor via an L298N driver. The system enables touch-free door control, improving safety and convenience in public and smart environments.**

**hardware connections for Arduino Uno-based Real-Time Distance Monitoring and Motor Speed Control** using:

* **Ultrasonic Sensor (HC-SR04)**
* **Motor Driver Module (L298N or L293D)**
* **DC Motor**
* **MATLAB-Arduino Serial Communication (via USB)**

**Hardware Components & Connections**

**1. Ultrasonic Sensor (HC-SR04) → Arduino Uno**

| **HC-SR04 Pin** | **Arduino Pin** |
| --- | --- |
| VCC | 5V |
| GND | GND |
| TRIG | D9 |
| ECHO | D10 |

**🔴 2. Motor Driver (L298N or L293D) → Arduino & DC Motor**

| **L298N Pin** | **Arduino Pin** | **Description** |
| --- | --- | --- |
| IN1 | D5 | Motor control input 1 |
| IN2 | D6 | Motor control input 2 |
| ENA (enable A) | D3 (PWM) | Speed control using PWM |
| OUT1, OUT2 | — | Connect to DC Motor terminals |
| VCC (Motor Power) | External 9–12V | Motor power (NOT Arduino 5V) |
| GND | GND | Common ground with Arduino |
| 5V (if available) | — | Optional: jumper for logic power |

If your motor needs more current, power L298N with an external 9–12V adapter.

**3. MATLAB Communication**

* **Connect Arduino Uno to PC via USB**
* Use MATLAB's serialport() or arduino() functions for communication.

**System Logic Overview**

* Arduino reads distance from HC-SR04.
* Sends distance via serial to MATLAB.
* MATLAB decides speed based on distance and sends a command back.
* Arduino changes motor speed via PWM (ENA) and direction via IN1/IN2.

Code :

% Initialize Arduino on COM5

arduinoObj = arduino('COM5', 'Uno', 'Libraries', 'Ultrasonic');

% Define ultrasonic sensor object (Trig = D9, Echo = D10)

ultrasonicSensor = ultrasonic(arduinoObj, 'D9', 'D10');

% Define motor control pins

motorIN1 = 'D5'; % IN1

motorIN2 = 'D6'; % IN2

motorENA = 'D3'; % ENA (PWM pin for speed)

% Set motor pins as digital outputs

configurePin(arduinoObj, motorIN1, 'DigitalOutput');

configurePin(arduinoObj, motorIN2, 'DigitalOutput');

configurePin(arduinoObj, motorENA, 'PWM'); % PWM for speed control

disp('System Initialized. Monitoring for object...');

prevState = 'none'; % To track last action and avoid repeating

while true

% Read distance and convert to cm

distance = readDistance(ultrasonicSensor) \* 100;

fprintf('Measured Distance: %.2f cm\n', distance);

if distance <= 50 && ~strcmp(prevState, 'open')

disp('Human detected: Opening the door (forward)...');

% Run motor forward

writeDigitalPin(arduinoObj, motorIN1, 1);

writeDigitalPin(arduinoObj, motorIN2, 0);

writePWMDutyCycle(arduinoObj, motorENA, 1); % Full speed

pause(2); % Open door for 2 seconds

% Stop motor

writeDigitalPin(arduinoObj, motorIN1, 0);

writeDigitalPin(arduinoObj, motorIN2, 0);

writePWMDutyCycle(arduinoObj, motorENA, 0);

disp('Door opened. Waiting for person to move away...');

prevState = 'open'; % Update state

elseif distance > 50 && ~strcmp(prevState, 'closed')

disp('No human nearby: Closing the door (reverse)...');

% Run motor in reverse

writeDigitalPin(arduinoObj, motorIN1, 0);

writeDigitalPin(arduinoObj, motorIN2, 1);

writePWMDutyCycle(arduinoObj, motorENA, 1); % Full speed

pause(2); % Close door for 2 seconds

% Stop motor

writeDigitalPin(arduinoObj, motorIN1, 0);

writeDigitalPin(arduinoObj, motorIN2, 0);

writePWMDutyCycle(arduinoObj, motorENA, 0);

disp('Door closed. Waiting for someone to arrive...');

prevState = 'closed'; % Update state

end

pause(0.3); % Sensor stabilization time

end